Please note:
That document needs update to the current state of the project. Some information might not be valid anymore, though the basics have not changed. We are working on a new document with updated and better structured information.

Arbitrator system

Version 0.9

- Arbitrator registration
  - Arbitrator collateral fund management
    - Limitation for standard tx
    - Bootstrapping
- Arbitrator selection
  - Attributes
  - Methods of conflict resolution
  - ID verification
- TLS Notary
- API
  - Register arbitrator
  - Get Arbitrator list by language
  - Call arbitrator for dispute resolution
  - Give reputation to arbitrator
  - Complain about arbitrator

The arbitration system will be a separate application connected with an open API to the exchange software.
That way the exchange should be less vulnerable if the arbitration system fails for some reason and might switch to an alternative arbitration system or include other 3rd party arbitrator systems which follow the API.

The arbitrators are responsible on their own for any legal and fiscal issues.
They can operate with their legal identity or anonymous.
The arbitrator system will be an open source software and there is no direct monetisation model connected to it (the fee goes directly from the trader to the arbitrator).
The funding for the development for the arbitration system will be covered by the model used for the exchange.
The data are stored in the same DHT network like the exchange.
The arbitration system will be decentralized, the only “centralized” entities are the individual arbitrators.
They are partitioned by language.

Arbitrator registration

To become an arbitrator the candidate need to fill in a standardized form.
He need to pay in a high collateral (e.g. 2-5 times more the the max. supported trade volume) which will be locked into a MultiSig address and will be returned when he wants to leave the arbitration pool. In case of an abuse of his arbitrator role (collusion with a trader) he will lose that collateral. It is planned that his confiscated collateral will be payed to the project donators.
Every arbitrator starts with an reputation of 1. If he lose his reputation he will also lose a part of his collateral.
In conflict cases the trader who called the arbitrator will give reputation feedback afterwards.
The collateral in the trade process will be set to the value of the arbitrators fee for active arbitration. The trader who calls the arbitrator has to pass the arbitrator a payout tx which contains the arbitration fee paid to the arbitrator and the changed payout to the traders. The collateral from the other peer will be used as payment for the arbitrator.
The arbitrator will only accept, sign and publish that payout tx if he can confirm the validity of the callers position. The contract of the trade will be passed to the arbitrator as well. As the contract is signed and its hash published in the blockchain, he does not need to verify the contract from the other peer.
The arbitrator might contact the other peer for more evidence in unclear cases.

If the trader is not satisfied with the arbitrators decision he can request another arbitrator from the pool. The selection will be done inside the pool and is based on reputation and experience (number of active cases) combined with a verifiable but unbiased selection model.
E.g.: (trader ID + arbitrator ID) % possible candidates. Candidates are the 10 arbitrators with the highest reputation for that particular language.
A given reputation from another arbitrator will count more than a traders reputation.
In severe cases the arbitrator can be kicked out and he loses his collateral.

The passive arbitration fee will be paid from the deposit tx in every trade.

Arbitrator collateral fund management

The collateral will be paid to a MultiSig address controlled by a number of other arbitrators.
The selection of those key holders will be done in a unbiased and verifiable way.
A similar model like above could be used.
The number of participants in that MS is limited because of tx size limitations.
Also it needs to be considered that arbitrators might disappear in an uncontrolled way.
When an arbitrator leave the pool and he is a key holder of a MS fund of another arbitrator, that fund will be moved to a new MS fund with new key holders.
In cases where an arbitrator has left without that procedure, the fund must not become inaccessible. Therefore the number of necessary signers must be smaller than n-2, where n is the number of participants.
E.g: A 4 of 6 MultiSig could be considered as safe enough.

Limitation for standard tx

Script size: 500 bytes
Size of a pubKey: 33-34 bytes
Size of sig: 74 bytes
Some extra bytes will be added as well.
4 of 6 MultiSig: 6*33 + 4* 74 = 494 bytes -> should be fine with a few extra bytes added.
Otherwise 3 of 5 would be acceptable as well.

Bootstrapping

It will start with an initial group of e.g. 5 arbitrators which have mutual trust relationship outside the system (personal contact, web of trust,...) The collateral fee can be lower at the beginning and be increased as soon as more arbitrators join.
To facilitate bootstrapping for a new language, the arbitrators from other languages could play roles for key management or conflict resolution.

Arbitrator selection

The offerer defines in his preferences a list of accepted arbitrators.
He selects those from an arbitrator list published by the arbitrator system provider.

Attributes

- Supported languages
- Nickname, Real name or company name
- Message ID
- Public key (used for 2-of-3 MultiSig)
- Arbitrator BTC address
- Arbitrators reputation
- Number of conflict resolutions
- Arbitrators country
- Fee for providing the service in % of trade volume (passive, no dispute)
- Fee for active arbitration in % of trade volume
- Min. Fee for active arbitration in BTC
- Method of conflict resolution
- Method of ID verification

Methods of conflict resolution

- TLS Notary
- Skype video chat
- Telephone call
- Support for legal interventions (if arbitrator is a lawyer)
- ...
- Other (to be described textual)

ID verification

- No ID verification needed
- ID verification with passport
- ID verification with other government issued ID
- ID verification with utility bill
- ID verification with social network ID (Facebook, G+, Twitter,...)
- PGP key
- Web of trust
- BTC OTC
- ...

The offers displayed in the orderbook are already filtered by the personal settings for the accepted arbitrators (the intersection set of both traders preferences will be the displayed offers).

That way there is no negotiation about the arbitration selection needed. The selection will happen in a verifiable way but there is no way to influence the selection by the trader.
An example for a possible solution would be:
The traders have 3 arbitrators in the intersection set (A1-A3).
(Trader_1_AccountID + Trader_2_AccountID) %3 = index of the selected arbitrator.
That way both can verify the selection and get the same result and no one can change the selection to his favor (preventing collusion with the arbitrator).

Update:
After discussion with waxwing we might consider other solutions like using a random number from http://www.nist.gov/itl/csd/ct/nist_beacon.cfm to prevent misuse of the selection mechanism.

TLS Notary

References:
https://github.com/tlsnotary/tlsnotary

The TLS notary solution can be used optional by arbitrators.
In the beginning there might be a trust and reputation problem for wider acceptance, so there should be arbitrators who are offering alternative resolution methods.
Mid and long term the TLS notary might serve as the preferred and most safe solution.

The TLS notary will be kept outside the scope of the exchange and arbitrator software. The arbitrators and the clients can use it as additional standalone tool.

API

The data will be stored in the DHT.
For collateral payment/refund a payment protocol will be used, similar to the trading protocol.

Register arbitrator

After the collateral payment an arbitrator get registered with all his supported languages.

void register(Arbitrator arbitrator, List<String> languages);

Get Arbitrator list by language

The exchange client will read the list of arbitrators from the DHT. That list is partitioned by the language.
TomP2P: locationKey = language ID
Inside that object there is a list of arbitrators keyed by the arbitrators messageID

List<Arbitrator> getAribtrators(String language);

Call arbitrator for dispute resolution

In case of a dispute the trader contact the arbitrator. There will be a defined answering time window until the arbitrator need to response (72h).
The message ID will be used for contacting him inside the messaging system.

Dispute:
- reason
- contract
- contact hours
- description
- pre signed payout tx

void resolveDispute(Arbitrator arbitrator, Dispute dispute);

Give reputation to arbitrator

After the arbitration the trader gives his rating.
That will be stored similar like the reputation for traders.
The arbitrator messageID is used as location key and inside that there is a list of ratings.
A trader can only put one rating in, but he might update a previous rating.
The rating will be signed by the trader and the messageID (pubKey) will be included as well, so the rating is verifiable by others.

Rating:
- satisfaction: 0-100%
- comment

void rateArbitrator(Arbitrator arbitrator, Rating rating)

Complain about arbitrator

In case that the trader is not satisfied with the arbitration he might make a complaint to the pool. He will be contacted by a selected member of the pool.

Complaint:
- dispute (from previous resolveDispute call)
- contact hours
- description

void reportComplaint(Arbitrator arbitrator, Complaint complaint);